Machine Learning for Computer Vision
Exercise 01
Tutor: Thorsten Beier

April 18, 2017

1 Rules

- Deadline for the exercise sheet is the next exercise class. Submit your results via email to `thorsten.beier@iwr.uni-heidelberg.de` **and** bring a printed version to the exercise class.
- The results should be submitted as a single zip file which should contain a single pdf answering the questions and showing the plots. Also include all your source code. The source code should have comments!
- Exercise can be done in groups of max. 2 persons.

2 Pixel Classification (15 Points)

As a first coding exercise we will implement a system to perform a pixel wise classification task. We will use the Weizmann Horse Database and train a system which predicts for each pixel if it is a background or horse pixel as shown in Figure 1.

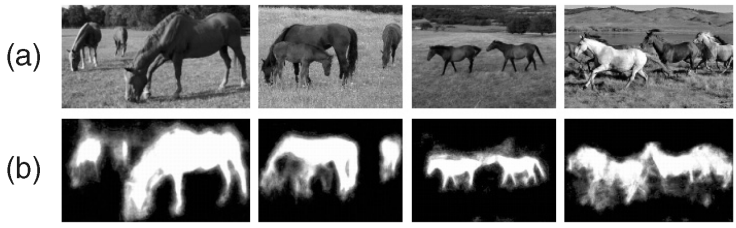

Figure 1: Pixel wise horse detection: In this exercise we will use RGB images instead of gray valued images. Image Credit: Auto-Context and Its Application to High-Level Vision Tasks and 3D Brain Image Segmentation

2.1 Read the Data (5 Points)

- Download the Horse RGB images `http://www.msri.org/people/members/eranb/rgb.tar.gz`
- Download the Horse figure-ground images `http://www.msri.org/people/members/eranb/figure_ground.tar.gz`
- Select a training set with 10 images of brown horses and a test set with 10 images brown horses.

2.2 Compute Features (5 Points)

- Compute different filters for each RGB horse image to create features for each pixel. Use filters from `http://scikit-image.org/docs/dev/api/skimage.filters.html`. You can use filters as 'gaussian','laplace' and 'median' filter. Compute the filters on each of the three channels (R,G and B). Vary the scale of the filters to generate more features.
- Plot the used filters. You can use matplotlib to show images `http://matplotlib.org/users/image_tutorial.html`.
- Compute these features for all images in the training and test set.

2.3 Random Forest (5 Points)

- Train a "Random Forest" classifier `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`. We use the computed filters as features, and the figure-ground images as training data.
- Predict the class memberships (horse vs. background) for all pixels for each RGB horse image in the test test. Plot the predictions and compare the predictions with the ground truth.

3 Graphical Models (5 Points)

Let $E(x_0, x_1, x_2)$ be an energy-function of 3 binary variables $x_0 \in \{0, 1\}$, $x_1 \in \{0, 1\}$ and $x_2 \in \{0, 1\}$. $E(x_0, x_1, x_2)$ is a sum of factors where each factor depends only on a subset of variables:

$$E(x_0, x_1, x_2) = \phi_0(x_0) + \phi_1(x_1) + \phi_2(x_1) + \phi_p(x_0, x_1) + \phi_p(x_0, x_2) + \phi_p(x_1, x_2)$$

The unary factors are defined as: $\phi_0(x_0) = \begin{cases} 0.1 & \text{if } x_0 = 0 \\ 0.9 & \text{if } x_0 = 1 \end{cases}$

$$\phi_1(x_1) = \begin{cases} 0.8 & \text{if } x_1 = 0 \\ 0.1 & \text{if } x_1 = 1 \end{cases} \qquad \phi_2(x_2) = \begin{cases} 0.9 & \text{if } x_2 = 0 \\ 0.1 & \text{if } x_2 = 1 \end{cases}$$

And the pairwise factor is a potts function: $\phi_p(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ 1 & \text{if } x_i \neq x_j \end{cases}$

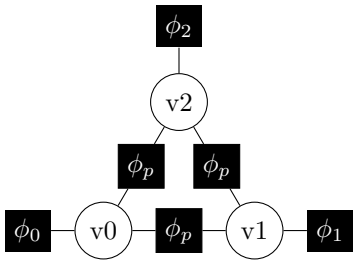

Figure 2: A graphical representation of $E(x_0, x_1, x_2)$ as Factor-graph. Each of the black boxes corresponds to a factor, each of the circle corresponds to a variable.

3.1 Minimize $E(x_0, x_1, x_2)$ (5Pt)

- Evaluate $E(x_0, x_1, x_2)$ by hand for all possible configuration of x_0, x_1 and x_2
- Which configuration of x_0, x_1 and x_2 minimizes $E(x_0, x_1, x_2)$? Comment on the result.